Measuring the effect of applying Differential Privacy on Machine Learning models

Abhay Krishna Arunachalam, Abhishek Vasudevan, Shantanu Agara Dwarakanath
CS 588 - Privacy in Networked and Distributed Systems
Department of Computer Science
University of Illinois at Chicago

1. Introduction

Privacy is a term used to describe an individual's anonymity and how safe they feel in a location. It can be called indistinguishability. Differential privacy is a statistical technique that aims to provide means to maximize the accuracy of queries from statistical databases while measuring (and, thereby, hopefully minimizing) the privacy impact on individuals whose information is in the database. The basic goal or aim of differential privacy can be summed up as "anything that can be learned about a respondent from the statistical database should be learnable without the access to the database." Differential Privacy ensures that the outcome of any analysis is essentially equally likely, and independent of whether any individual joins, or refrains from joining, the dataset.

Roughly, an algorithm is differentially private if an observer seeing its output cannot tell if a particular individual's information was used in the computation. Differential privacy is often discussed in the context of identifying individuals whose information may be in a database. Although it does not directly refer to identification and reidentification attacks, differentially private algorithms probably resist such attacks.

A technique introduced by Cinthia Dwork[1], differential privacy (DP) protects user privacy by adding random noise to the data. For example, adding three years to your age while answering a survey protects your information. As the data provided is noisy, it cannot be used to identify a user.

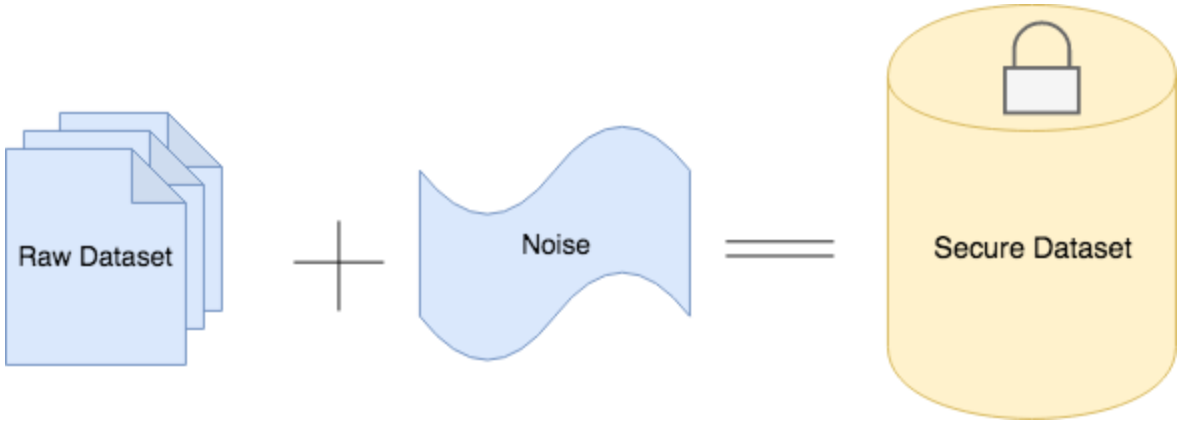

Fig 1.1 Differential Privacy, Image Credits: Abhishek Tandon

As data scientists it is of utmost importance to be extremely careful when working with datasets that contain sensitive information. Often, people dealing with these datasets equate privacy to removing names, SSNs, ID's, and credit card numbers. Unfortunately, this does not eliminate the privacy risk completely. While removing direct identifiers can help, there are more information elements in a dataset that can be used to re-identify an individual. For example, a Harvard study[2] from a couple of years ago proved that 87 percent of the US population can be re-identified using zip code, gender, and date of birth. With this project, we attempt at effectively reducing the privacy risk of a dataset

while maintaining its analytical value for ML. This is termed Privacy-preserving Data Mining (PPDM).

To summarize, Differential privacy is a system for sharing useful statistical information about datasets while, at the same time, withholding sensitive information so that not a particular record or group of records cannot be identified in the dataset. This is done by applying generalization and suppression techniques to certain points in the dataset to either eliminate them as outliers or group them into broader classes so that their specificity and hence “identifying potential” is decreased.

2. Analysis

To reduce the privacy risk of a dataset we will be analyzing information loss in a synthetic dataset with Logistic Regression, Decision Tree Learning, Support Vector Machine and k-nearest neighbors. We will be comparing how these four machine learning models fare when pit against each other and we will conduct an empirical analysis. A brief introduction to these algorithms is as follows:

2.1 Logistic Regression

Logistic regression[3] is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2.2 Decision Tree learning

Decision tree learning[4] uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

2.3 Support Vector Machine

Support vector machines[5] (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

2.4 k-nearest Neighbors

The k-nearest neighbors algorithm (k-NN)[6] is a non-parametric method used for classification and regression. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

3. Dataset

The synthetic dataset used in this project is originally adapted from Kaggle[7] with some minor tweaks to introduce a sensitive data field as well as remove some extraneous features. The synthetic dataset contains the following features and each feature has a privacy attribute as shown in right:

SSN	Identifying
Sex	Quasi-identifying
Age	Quasi-identifying
Race	Quasi-identifying
Marital Status	Quasi-identifying
Education	Quasi-identifying
Native Country	Quasi-identifying
Work Class	Quasi-identifying
Occupation	Quasi-identifying
Salary Class	Insensitive

It is important to note that the Social Security Numbers(SSN) in this data set were artificially generated by us using a script. The explanation of the privacy attributes are as follows:

I. **Sensitive**: No risk of identification from this data, but when the individual is identified, this data would reveal personal information about the individual.
II. **Insensitive**: No risk of identification and does not damage the individual in any way.
III. **Identifying**: Data which can be used on their own to identify individuals
IV. **Quasi-identifying**: Data which on their own do not reveal much, but when used with other data pose a risk of revealing identifying information.

4. Implementation and Working

We achieved the k-anonymity property to implement differential privacy. k-anonymity is a property possessed by certain anonymized data. The concept of k-anonymity was first introduced by Latanya Sweeney and Pierangela Samarati in a paper published in 1998[8] as an attempt to solve the problem: "Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful." A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least k - 1 individuals whose information also appears in the release.

The k-anonymity algorithm works on two main principles - **generalization** and **suppression**. Generalization refers to the process of decreasing the specificity of data by making it a part of a broader class, from which distinguishing the data points becomes increasingly difficult for an external attacker. This technique is implemented by providing several hierarchies or levels of generalization, which correspond to increasingly broader classification of the feature's values. For example, the individual data points of Color feature, [Red, Blue, Green, Yellow, Orange, White, Black] may be generalized into Warm Colors = [Red, Yellow, Orange], Cool Colors = [Blue, Green] and Neutral Colors = [Black, White]. This leads to some information loss, as this classification does not reveal the underlying individual datapoint, but at the same time, prevents an attacker from identifying the actual datapoint.

Suppression refers to the process of removing data from the dataset by replacing it with a null character(- or *). This technique is applied to outliers or to points which are difficult to generalize. It is important to remove outliers as they increase the entropy of the information in the database, leading to more knowledge gained by the attacker. The goal is for outliers not to be individually distinguishable. Their identification may expose valuable quasi-identifiers which can be leveraged to further decrease the privacy of the dataset. The degree of suppression is controlled by a suppression limit, which specifies the maximum proportion of records that the algorithm can suppress.

We tested our synthetic dataset against increasing values of k and made sure that each record should not be distinguishable from k other records. We developed several levels of generalization hierarchies depending on to what extent the dataset is to be generalized, as well as to comply with higher values of k. These hierarchies fall into two categories - Order Hierarchy and Data Hierarchy. Order Hierarchy is a type of generalization technique applied to numerical or integral features, where a continuous range of values is divided into groups. For instance, the age hierarchy may be in range of [0-10], [0-25], [0-50] and so on. On the other hand, Data Hierarchy is applied on categorical values to provide layers of abstraction depending upon k parameter. These hierarchies are utilized by the protection algorithm to suitably generalize the data or remove rows as necessary. The extent to which these techniques are applied depends upon the value of k. After this privacy was applied, the performance of all ML algorithms on both unprotected and protected data was studied, with “Salary Class” as a dependent feature. For fairness, the same parameters were maintained for a particular classifier for both the unprotected and protected dataset.The general hypothesis was an increase in information loss, and hence decrease in accuracy score of the classifiers.

5. Evaluation

As expected, we observed an escalation in information loss with an increase in K value. The same effect was perceived in the performance of the classification algorithms

- After PPDM techniques were applied, the training phase suffered from lack of complete information
- One-hot vectors corresponding to dependent, independent features represented misinformation, as they would have become sparse due to suppression.

5.1 Dataset comparison for different values of k

Figure 5.1.1 and 5.1.2 shows snapshots of the dataset when the value of K=5 and when it is 25. It is clear from here that anonymizing the dataset with lower values of K yields a dataset where more data points are unaffected. The asterisk (*) represents a data point which has been suppressed. Note that there are more asterisks when K=25 indicating that more data has been suppressed from the dataset.

ssn	sex	age	race	marital_status	education	native_country	workclass	occupation	salary_class
*	Male	[37, 56]	White	Never-married	Higher Education	North America	Government	Other	<=50K
*	Male	[37, 56]	White	Married-civ-spouse	Higher Education	North America	Non-government	Non-technical	<=50K
*	Male	[37, 56]	White	Divorced	Secondary Education	North America	Non-government	Non-technical	<=50K
*	Male	[37, 56]	Black	Married-civ-spouse	Secondary Education	North America	Non-government	Non-technical	<=50K
*	*	*	*	*	*	*	*	*	*
*	Female	[37, 56]	White	Married-civ-spouse	Higher Education	North America	Non-government	Non-technical	<=50K
*	Female	[37, 56]	Black	Married-spouse-absent	Secondary Education	North America	Non-government	Other	<=50K
*	Male	[37, 56]	White	Married-civ-spouse	Secondary Education	North America	Non-government	Non-technical	>50K
*	Female	[17, 36]	White	Never-married	Higher Education	North America	Non-government	Technical	>50K
*	Male	[37, 56]	White	Married-civ-spouse	Higher Education	North America	Non-government	Non-technical	>50K

Figure 5.1.1 - Snapshot of dataset when K=5

ssn	sex	age	race	marital_status	education	native_country	workclass	occupation	salary_class
*	Male	[37, 56]	White	Never-married	*	North America	Government	*	<=50K
*	Male	[37, 56]	White	Married-civ-spouse	*	North America	Non-government	*	<=50K
*	Male	[37, 56]	White	Divorced	*	North America	Non-government	*	<=50K
*	Male	[37, 56]	Black	Married-civ-spouse	*	North America	Non-government	*	<=50K
*	Female	[17, 36]	Black	Married-civ-spouse	*	North America	Non-government	*	<=50K
*	Female	[37, 56]	White	Married-civ-spouse	*	North America	Non-government	*	<=50K
*	*	*	*	*	*	*	*	*	*
*	Male	[37, 56]	White	Married-civ-spouse	*	North America	Non-government	*	>50K
*	Female	[17, 36]	White	Never-married	*	North America	Non-government	*	>50K
*	Male	[37, 56]	White	Married-civ-spouse	*	North America	Non-government	*	>50K

Figure 5.1.2 - Figure 5.1.1 - Snapshot of dataset when K=25

Figures 5.1.3 shows the correlation between information loss and value of K. For higher values of K, there is more information loss. For K values near 10000, the information loss is almost 80% in our dataset. Although having such a K value is impractical in our dataset because most of the dataset is lost, it proves to show the effects of the anonymization algorithm with higher K values.

Figure 5.1.4 shows correlation between information loss and risk factor which is returned by our module. It is clear that a lower information loss helps a potential attacker to extract more information and thereby increase the risk factor and vice-versa. Hence, information loss is inversely correlated to risk factor.

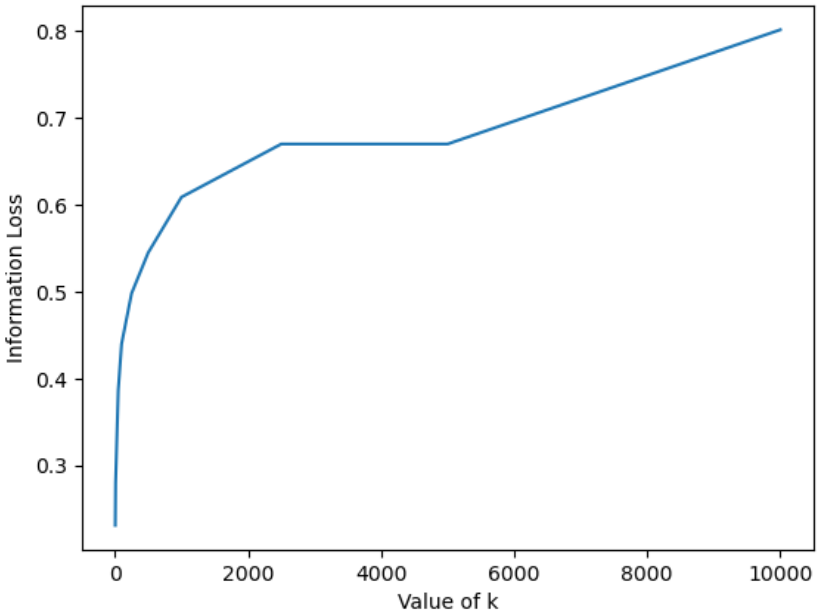

Figure 5.1.3 - Graph showing correlation between K value and Information loss

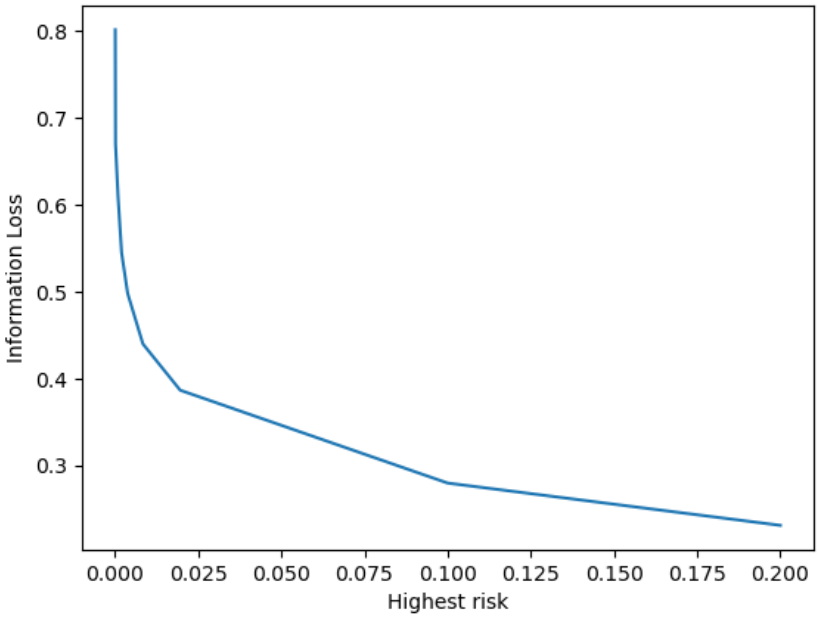

Figure 5.1.3 - Graph showing correlation between higher risk and information loss

5.2 Impact on various classification algorithms

We test four different algorithms namely: Logistic Regression, Decision Tree, Support Vector Machines and K-nearest neighbors. The idea is to evaluate the accuracy of the

algorithms on the unprotected dataset and then compare it to how the accuracy changes after the dataset has been protected. We train the algorithms by keeping the salary class as the output class which has to be predicted. On the unprotected dataset, we see all the algorithms have performed fairly well except Support Vector Machine which has a low accuracy as shown in Figure 5.2.1.

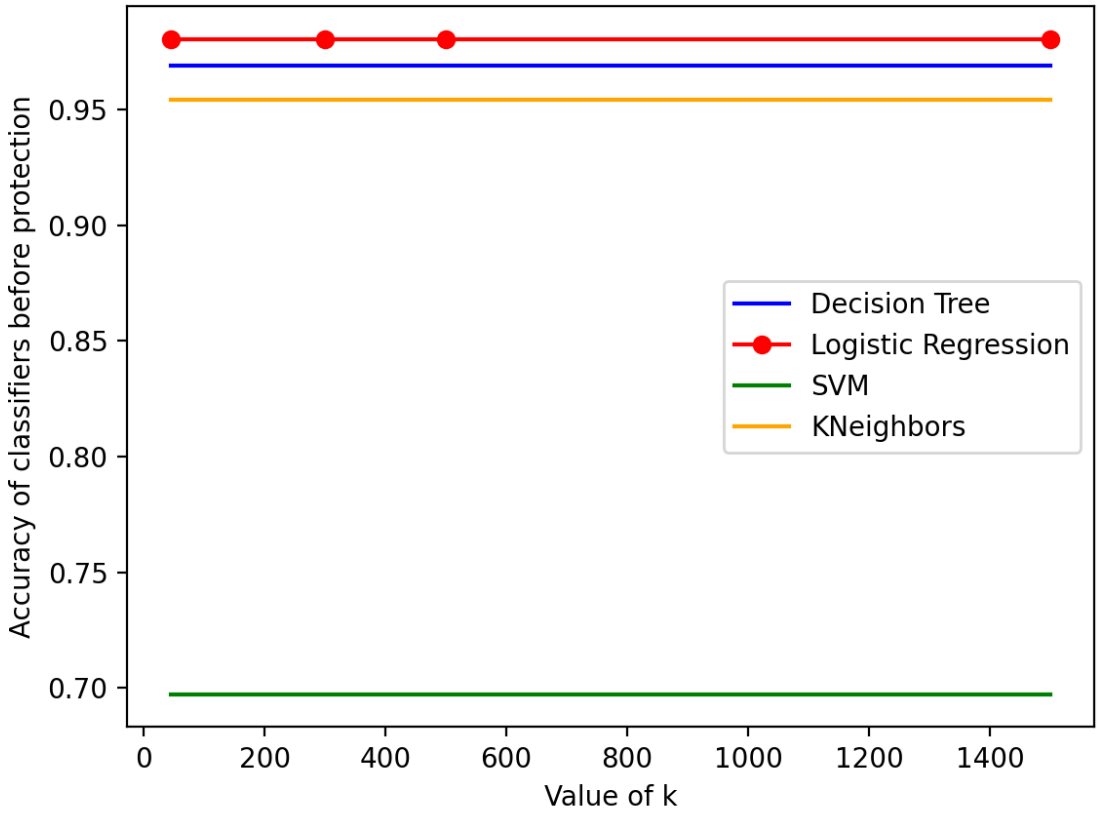

Figure 5.2.1 - Performance of algorithms on Unprotected dataset

Figure 5.2.2 shows the performance of the algorithms with varying K values applied for the K-anonymization algorithm on the dataset. The first thing to notice is that all the algorithms have decreased accuracy compared to the accuracy on the unprotected dataset which is expected. The only exception is the Support Vector Machine which has shown a 10% increase in accuracy when K = 5, although this decreases with higher K values. Such an increase in performance of classification after anonymization was also observed in [9]. Both SVM and Decision Tree perform almost the same with K-neighbors having a lower accuracy. The similarity in performance of the algorithms could be attributed to the fact that the dataset loses non-linearity after the dataset has been anonymized. Also, due to this reason, all algorithms show a slight increase in accuracy as K increases indicating a decrease in non-linearity. In practice, the choice of K value should be based on how sensitive the information is. The appropriate K value should be chosen considering the trade-offs between information loss and accuracy of the algorithms.

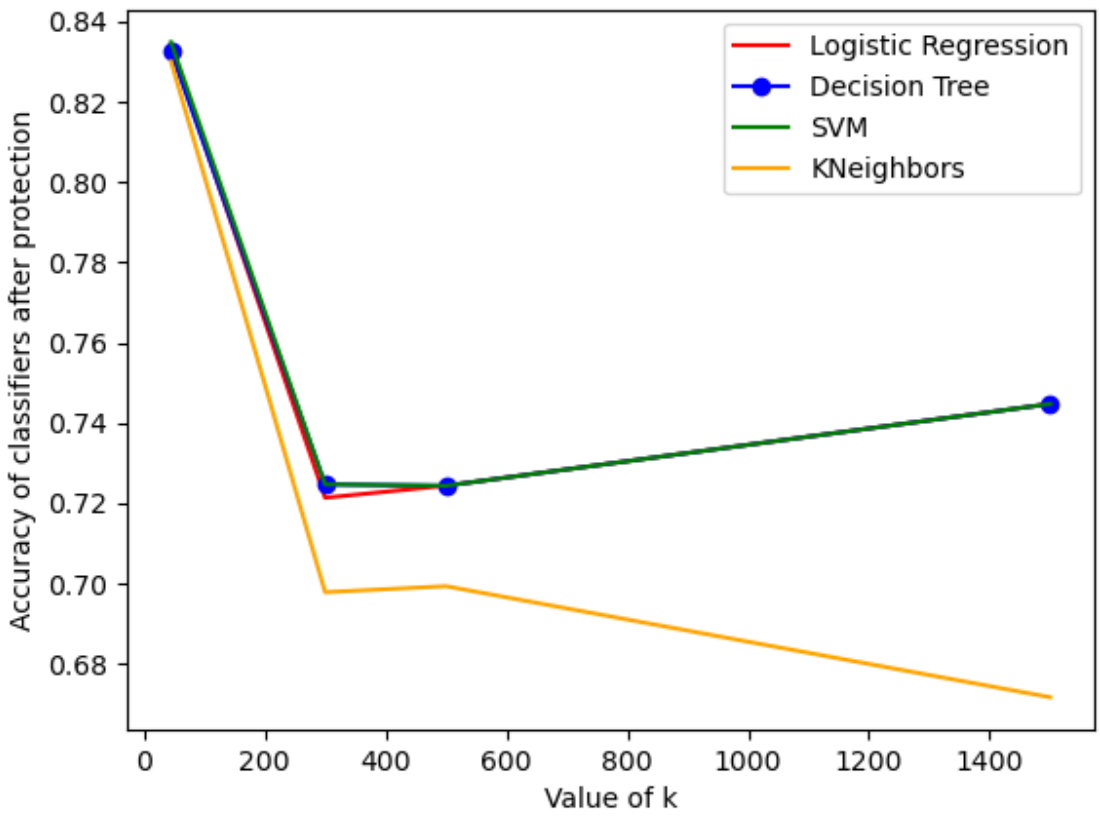

Figure 5.2.2 - Performance of algorithms on Protected dataset with varying value of K

6. GitHub

The source code of our project is available on GitHub at [10] with a README file explaining how to execute the code.

7. References

[1] C. Dwork, "Differential privacy," in Automata, languages and programming, ed: Springer, 2006, pp. 1-12
[2]L. Sweeney. k-anonymity: A model for protecting privacy. International J. of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557{570, 2002
[3] “Logistic Regression”, Wikipedia, Wikimedia Foundation
[4] “Decision Tree Learning”, Wikipedia, Wikimedia Foundation
[5] “Support Vector Machines”, Wikipedia, Wikimedia Foundation
[6] “k-nearest neighbors”, Wikipedia, Wikimedia Foundation
[7] https://www.kaggle.com/uciml/adult-census-income#adult.csv
[8] P. Samarati, L. Sweeney, “Generalizing data to provide anonymity when disclosing information”, PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, May 1998
[9] H.Wimmer, L.Powell, “A comparison of the effects of K-anonymity on Machine learning algorithms, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 11, 2014
[10] GitHub repository: https://github.com/abhishek-v/CS_588